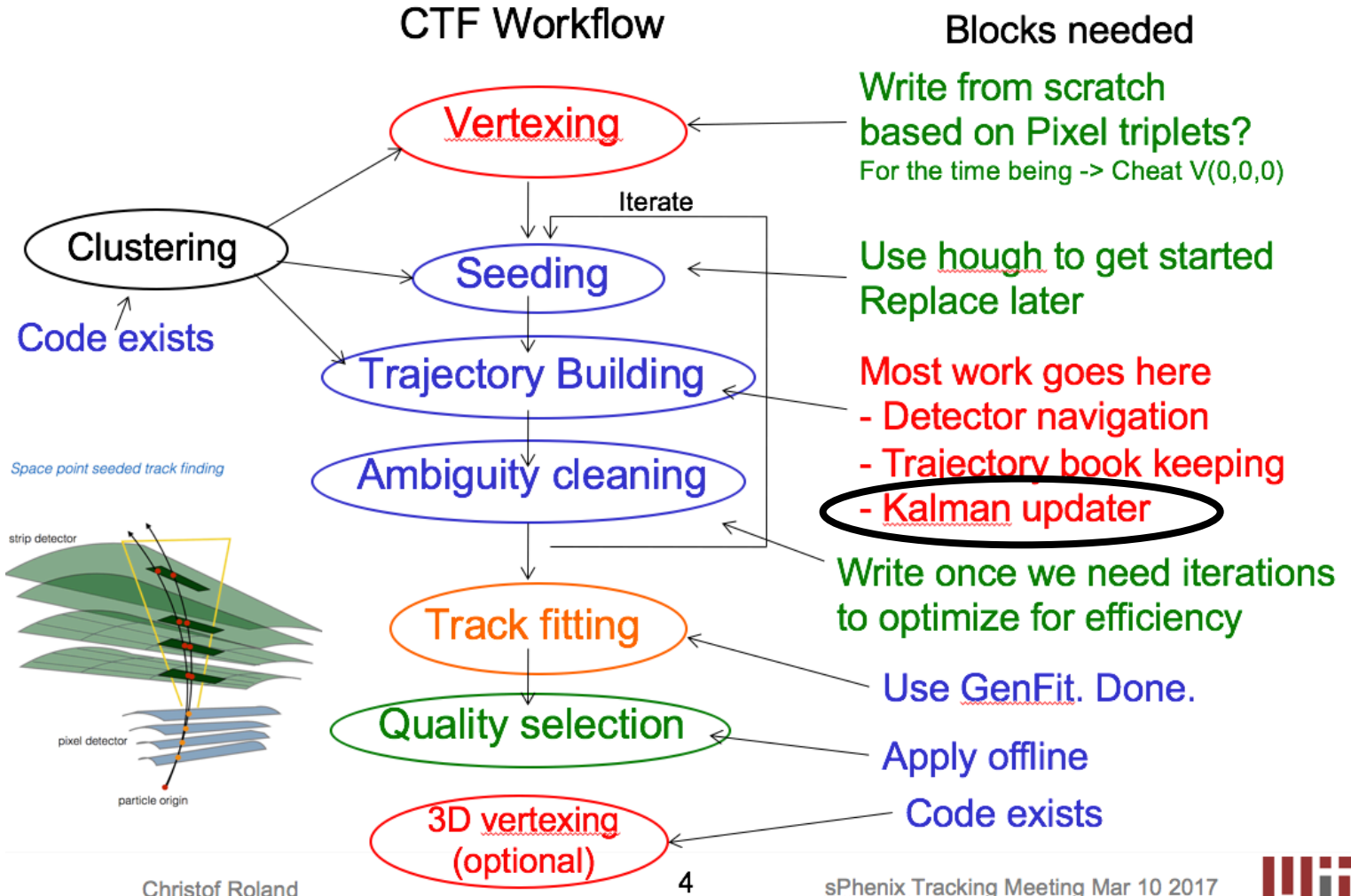
Kalman Updater

Jin Huang(BNL), Christof Roland(MIT), Haiwang Yu (NMSU)

Christof's proposal

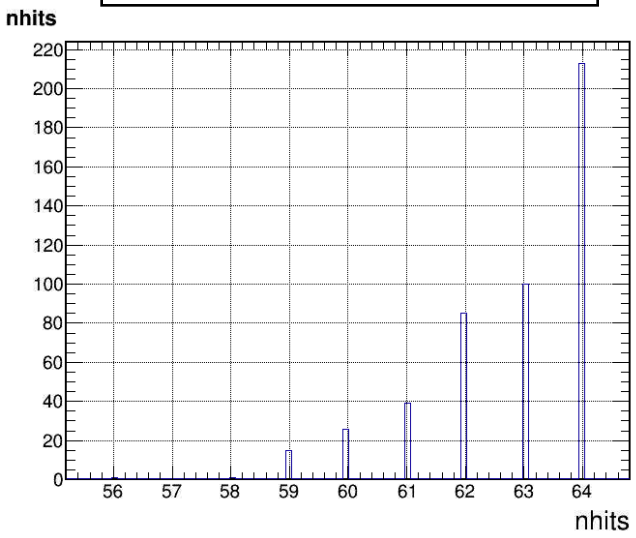
Building blocks for a combinatorial track finder (CTF)



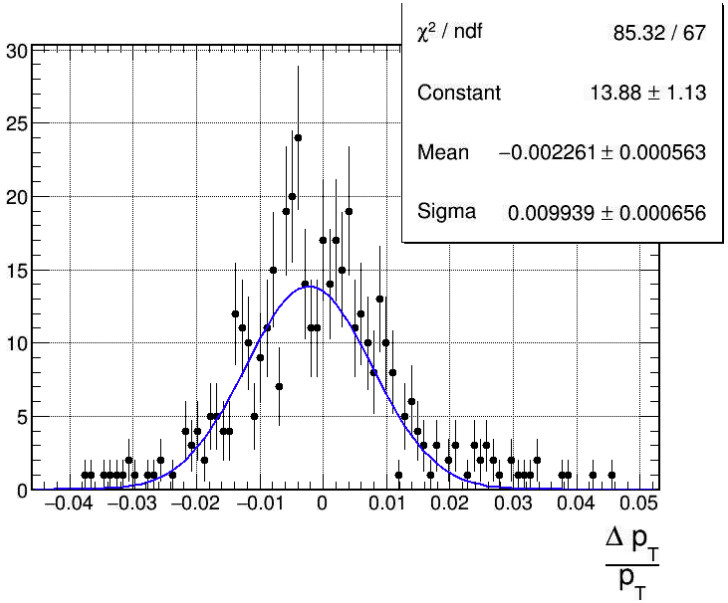
Kalman Updater

- PHGenFit::Track::updateOneMeasurementKalman
  - Spitting out tracks
- Stored cluster ID in *PHGenFit::Measurement* and *PHGenFit::Track*
- Test bench setup
  - PHG4KalmanPatRec
- Tested with single 10GeV pi+
  - cylindrical MVTX+IT+TPC
  - silicon seeding

Picking up most hits.
Missing 3 layers, will dig it out

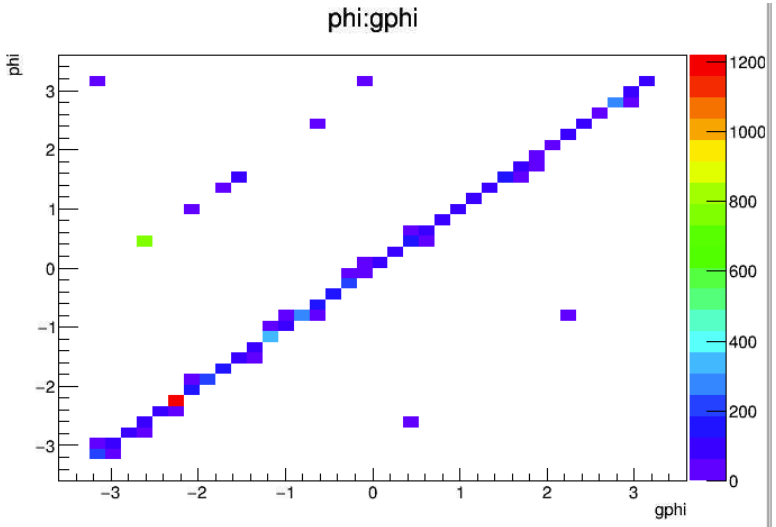


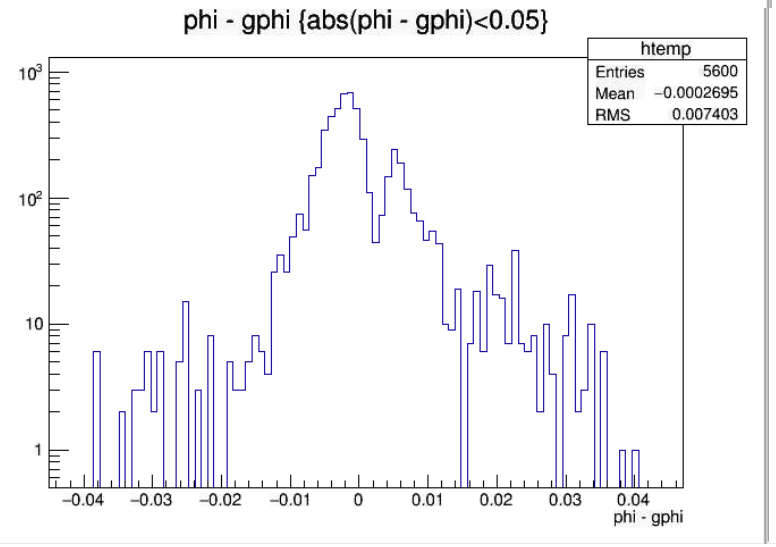
Momentum resolution looks nice



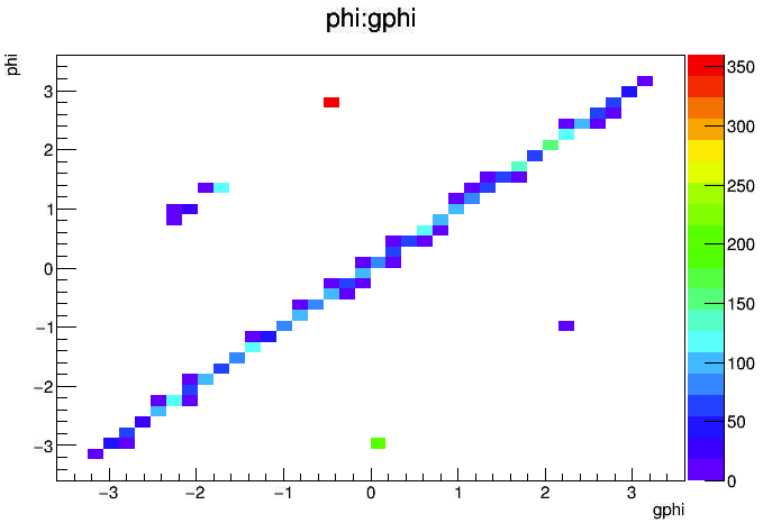
Seeding: $\Delta\phi$

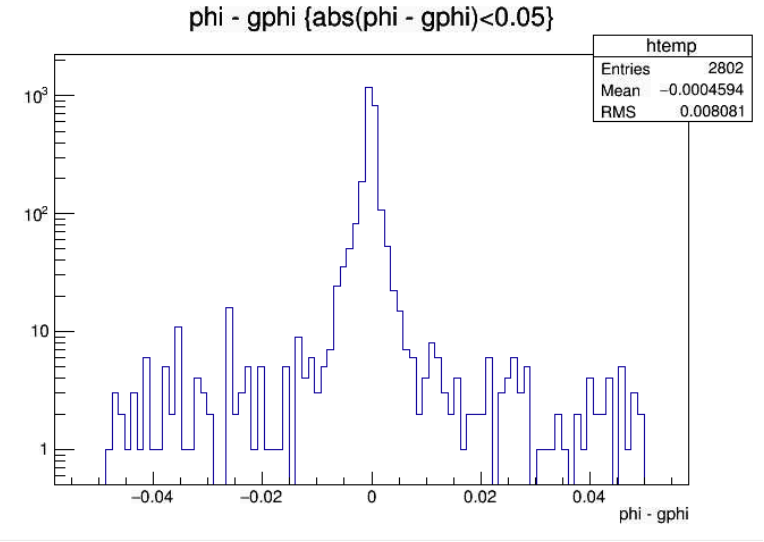
Silicon
layer: 0, 1, 2, 3, 4, 5, 6





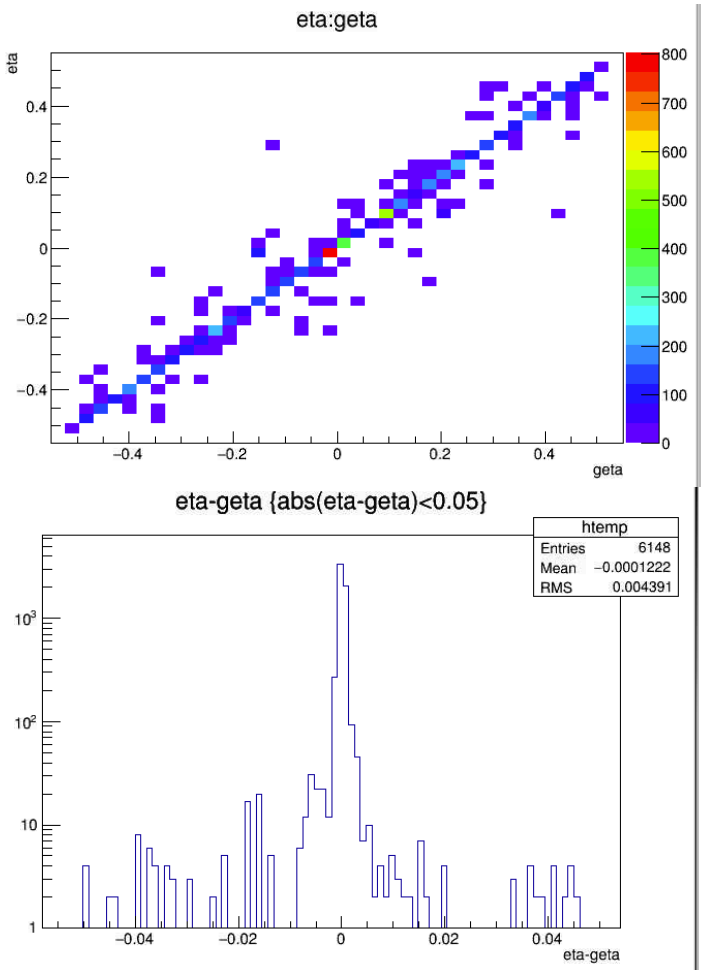
part Silicon + first TPC
layer: 0, 1, 2, 3, 5, 7



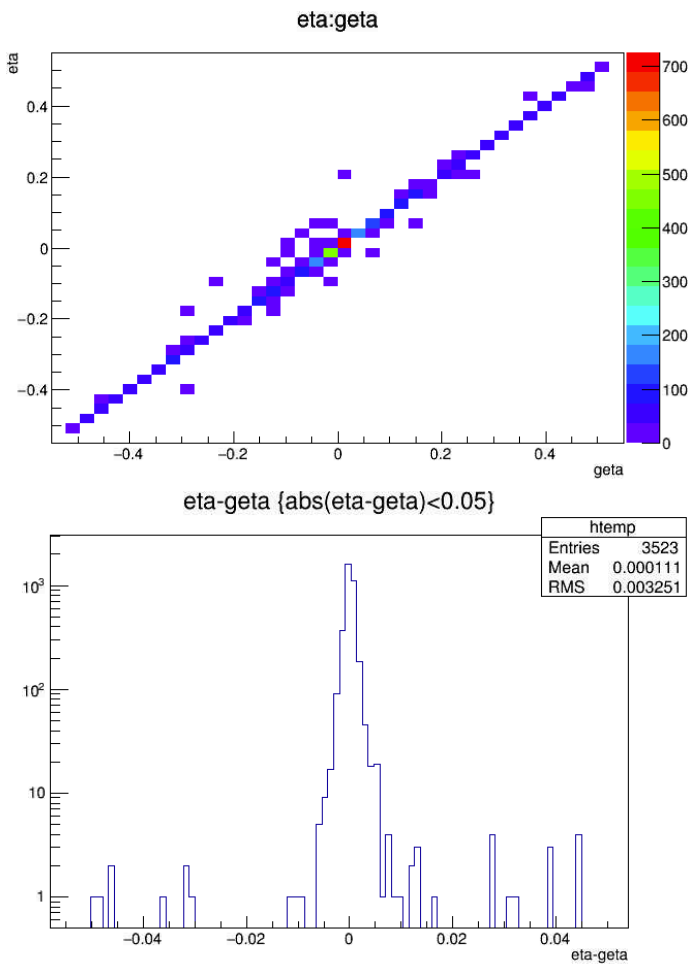


Seeding: $\Delta\eta$

Silicon
layer: 0, 1, 2, 3, 4, 5, 6

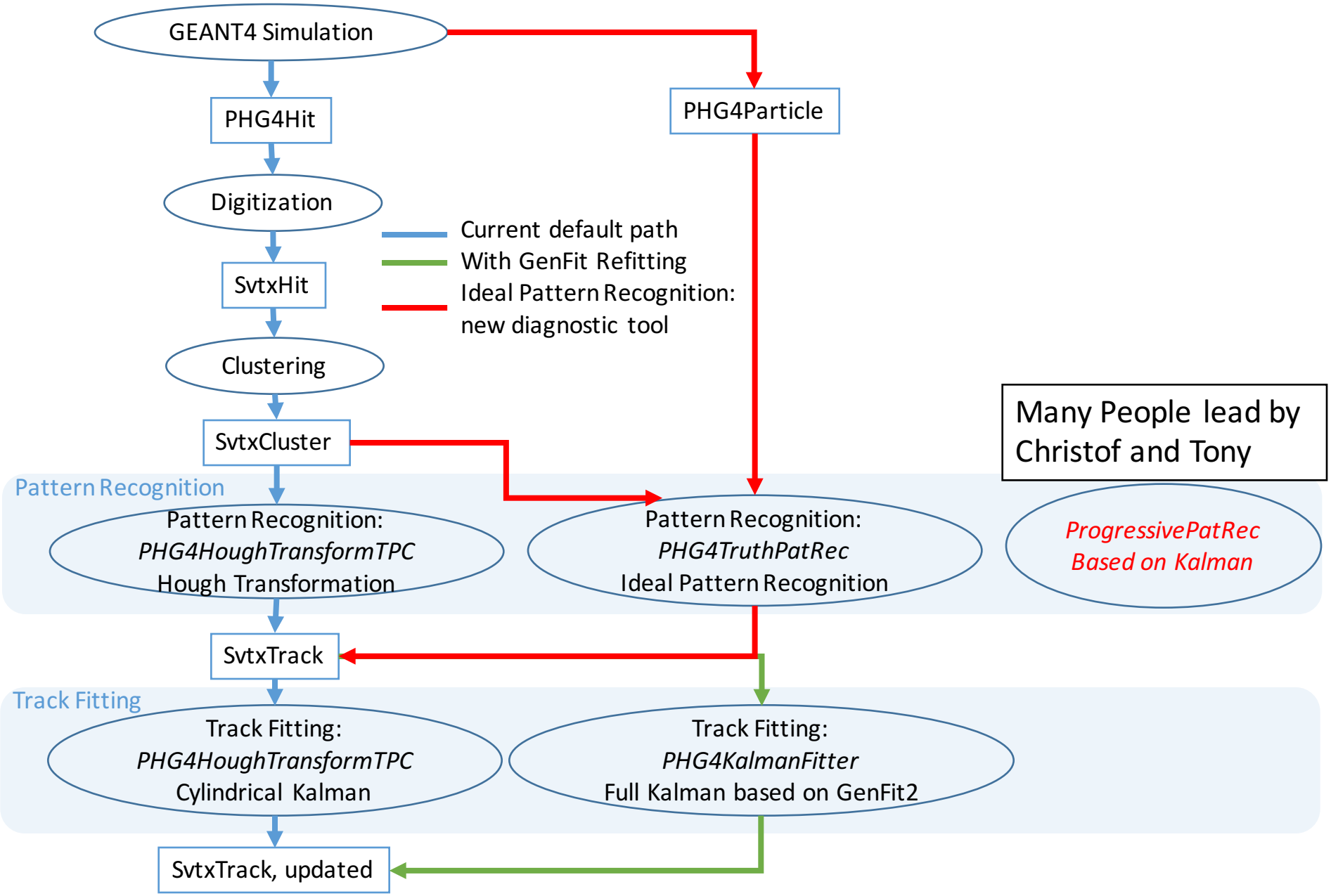


part Silicon + first TPC
layer: 0, 1, 2, 3, 5, 7



Backups

sPHENIX tracking



Track updater with one measurement

PHGenFit::Track

genfit:::Track

vector::<AbsTrackRep>

TVectorD stateSeed_; // 6D: position, omentum TMatrixDSym covSeed_; // 6D

Only affect the starting TrackPoint => trackPointID == 0

vector::<TrackPoint>
= > map< AbsTrackRep*, AbsFitterInfo* >

Where measurements and KalmanFitterInfo resides

maps::<AbsTrackRep*, FitStatus>

Fitting Status: converged? chi2?

new

updateOneMeasurementKalman(
vector<PHGenFit::Measurement*>& measurements,
map<double, PHGenFit::Track*>& incr_chi2s_new_tracks
)

return a map incr_chi2 ⟹ new Track

update

genfit::MeasuredStateOnPlane* extrapolateToCylinder(…)

extrapolation now works with tracks without fitting